Remerciements

Je voudrais remercier, M. Stéphane Laurent, le responsable de la cellule bureautique pour m'avoir accepté dans ce stage, et pour avoir été disponible tout au long de cette période, ainsi que M. Bernard Mayer, membre de l'équipe bureautique, qui fût mon maitre dans le domaine du PHP.

Je tiens ensuite à remercier tout le personnel de la Direction Informatique et Logistique qui a su m'accueillir dans de très bonnes conditions, et plus particulièrement le personnel de la cellule bureautique, qui a su m'intégrer dans l'équipe de développement et ainsi, me donner tous les outils et toute l'aide nécessaire au bon déroulement de ce projet.

Enfin, je remercie l'intégralité des enseignants de l'IUT, qui on participé à ma formation, et notamment mon tuteur de stage M. Saïd Fettahi.

CRÉDIT AGRICOLE

Sommaire

1 PRÉSENTATION DE L'ENTREPRISE 4

1.1 Présentation du groupe Crédit Agricole 4
1.2 La communauté Comète 6
1.3 La cellule bureautique 7

2 THÈME ET ACTIVITÉS 8

2.1 Présentation du sujet 8
2.2 Cahier des charges : 8
2.3 Contact humain : 10
2.4 Moyens matériels et logiciels 11
2.5 Méthodes de travail : 13

3 DÉVELOPPEMENT DE L'APPLICATION : 14

3.1 L'analyse de la base de données ACCESS : 14
3.2 Le MCD : Modèle Conceptuel de Données : 16
3.3 Gérer les bases de données sous les outils MySQL 20
3.4 La phase développement PHP : 23

4 RÉFLEXION : 30

4.1 Résultat, la théorie et la pratique : 30
4.2 Le rôle de l'utilisatrice : 31

5 CHRONOLOGIE DU STAGE 32

CONCLUSION: 33

GLOSSAIRE 34

SOURCES DOCUMENTAIRES 37

Annexe 1: Modèle de lettre à accomplir
Annexe 2 : Exemple d'une requête SQL
Annexe 3 : Exemple de détection d'erreur:
Annexe 4 : Plan de l'application

CRÉDIT AGRICOLE

Introduction :

Dans le cadre de mes études en I.U.T Informatique j'ai effectué mon stage de fin d'année d'une durée de dix semaines au sein de la cellule bureautique de la D.I.L. (Direction Informatique et Logistique) de la Caisse Régionale du Crédit Agricole Charente-Périgord.

A mon arrivée, il n'existait pas d'outil de suivi des prêts in fine arrivant à échéance. Il s'agit de prêts sur des clients NEIERTZ* (surendettés). Ces crédits doivent être régularisés soit par la mise en place d'un nouveau plan BDF, soit par une rentrée prévue d'argent.
Cependant, il arrive fréquemment que les clients oublient l'arrivée à échéance de leur crédit et, donc, n'anticipent pas les modalités de son remboursement. Le prêt est alors identifié lors de son passage en CDL (Crédit Douteux Litigieux), ce qui génère des dysfonctionnements pour sa régularisation.
Pour limiter ces cas d'anomalie, une demande de projet est créée pour prévenir ces clients trois mois avant l'échéance de leur prêt.

Dans un premier temps, je ferai une présentation du Groupe Crédit Agricole et plus particulièrement du Système d'Information Comète dont dépend C.A.C.P. (Crédit Agricole Charente-Périgord). Ensuite, j'expliquerai le sujet et je présenterai les activités. Puis, seront exposées et détaillées les phases du développement du projet ainsi que les difficultés rencontrées. Suivront la phase de test et de validation du projet.

NEIERTZ : La loi «NEIERTZ" a été adoptée le 31 décembre 1989, elle a instauré un dispositif de règlement collectif des dettes des particuliers surendettés... (Voir le glossaire pour les détails)

1 Présentation de l'entreprise

1.1 Présentation du groupe Crédit Agricole

Le Groupe Crédit Agricole, premier groupe bancaire de France, figure également aux premiers rangs mondiaux, si l'on en juge l'importance de ses fonds propres.
L'organisation du Crédit Agricole est basée sur une architecture à trois niveaux : Les 2 583 Caisses Locales sont regroupées en 41 Caisses Régionales et disposent d'un réseau de 7 142 agences et de 11 300 automates bancaires.

Pour renforcer leur solidité et leur compétitivité afin d'assurer le meilleur service à leurs clients, les Caisses Régionales se regroupent. Depuis 1988, leur nombre a ainsi été ramené de 94 à 41 (à fin août 2005). Chaque fusion donne lieu à un processus minutieux de préparation permettant au Crédit Agricole de conserver son ancrage sur le terrain et son atout de banque de proximité.

Le Crédit Agricole a construit une dimension véritablement européenne en s'appuyant sur les implantations de ses principales filiales, notamment dans le crédit à la consommation, la gestion d'actifs et la gestion privée, et en nouant un réseau d'alliances et de partenariats stratégiques avec de grandes banques européennes.

Les Caisses Régionales gèrent la majorité du capital du Crédit Agricole S.A. (CASA), qui est le cœur du réseau bancaire Crédit Agricole. Cette organisation décentralisée permet un service de proximité omniprésent, très largement apprécié par plus de 16 millions de clients.

En tant qu'organe central, le Crédit Agricole S.A. :

- Veille à la cohésion et au bon fonctionnement du réseau, ainsi qu'au respect des normes garantissant la liquidité et la solvabilité des Caisses Régionales ;
- Banque centrale, il garantit l'unité financière du Groupe ;
- Enfin, il a la charge de la cohérence de son développement.

Histoire du Groupe Crédit Agricole :

A la fin du XIXe siècle, la création du Crédit Agricole répond à la nécessité de doter l'agriculture d'un outil de financement propre avec des crédits plus longs et plus souples. Pragmatique, le CA a édifié son modèle à partir de la base sous une forme mutualiste avec le soutien des pouvoirs publics. Indépendant financièrement en 1967, il s'émancipe de la tutelle de l'Etat en 1988.

Présentation du Crédit Agricole Charente-Perigord :

116 agences bancaires, réparties sur les deux départements, une Banque Privée, 2 centres d'affaires entreprises, 2 espaces conseil habitat, et la banque par téléphone lui permettent d'avoir une relation de proximité avec ses 245 000 clients. Cette clientèle est répartie en quatre marchés : les particuliers, les professionnels et exploitants agricoles, les entreprises, les coopératives et les collectivités locales. Les caisses régionales de Charente et de Dordogne ont été respectivement créées en 1899 et 1904, leur fusion, quant à elle n'a été faite que beaucoup plus tard, en 1994.

1.2 La communauté Comète

Le GIE Comète :

D'un point de vue informatique, le Crédit Agricole référence aujourd'hui 5 systèmes d'information (S.I.). Chaque Caisse Régionale est affiliée à l'un de ces SI. Parmi ces Systèmes d'Information, l'un d'entre eux, Comète, regroupe les cinq Caisses Régionales suivantes : Charente-Maritime / Deux-Sèvres (CMDS), Normandie-Seine (CRNS), Oise (CR60), Nord de France (CRNF) et Charente - Périgord (CRCP). Aujourd'hui cette répartition est en train de changer, car l'Oise quitter le groupe comète pour rejoindre un autre SI du crédit agricole.

Présentation de Comète :

Le GIE Comète (Groupement d'Intérêt Economique) a été créé en 1997. Cette communauté est en phase avec les directives nationales tant au plan de l'offre client qu'au plan de l'intégration des logiciels nationaux. L'architecture relativement récente de Comète permet une grande indépendance du système d'information par rapport aux organisations et notamment dans la localisation des fonctions siège, unité de gestion et agence. Les fonctionnalités couvertes par Comète sont tournées vers la vente, la gestion du risque, et l'optimisation des traitements de masse.

Après des années consacrées à la réalisation des migrations, le GIE trouve une certaine stabilité (à partir de 1999) tant sur le plan des modes de fonctionnement, que des coûts informatiques et des fonctionnalités.
Aujourd'hui, Comète continue à prendre en compte les attentes du marché bancaire dans tous les domaines, favorisant le service à la clientèle.

Une organisation répartie :

Depuis sa création, le GIE a organisé la répartition des responsabilités et des compétences pour l'évolution et la maintenance du système d'information Comète. Chaque caisse de la communauté Comète s'occupe d'une partie du SI qui lui a été déléguée, elle anime un pôle fonctionnel au niveau de l'expression des besoins et la production de cahiers des charges. Cette répartition se veut équilibrée entre les différents membres de la communauté et évoluera en cas d'arrivée d'un nouveau membre.

La communauté Comète ne compte pas moins de 110 développeurs, répartis entre les équipes des différentes caisses.

COMETE, une communauté en mouvement :

Mouvement d'élargissement : depuis sa création, la communauté COMETE n'a cessé de grandir, ce qui reflète la qualité de son Système d'Information.
Mouvement d'adaptation : la communauté a réussi à s'adapter à un nouveau système d'exploitation et a su faire face à l'intégration de nouveaux entrants.
Mouvement de développement : des nouvelles versions sont disponibles régulièrement.
Mouvement d'harmonisation : toutes les caisses régionales de la communauté convergent fortement sur les mêmes fonctionnalités.

1.3 La cellule bureautique

La cellule bureautique fait partie intégrante du service développement dont la mission est d'optimiser et d'assurer le traitement et la restitution de l'information sous toutes ses formes, aux clients internes comme externes, dans les délais impartis. Cette équipe de quatre développeurs d'applications a pour but d'analyser les besoins des utilisateurs puis de créer ou de modifier des applications pour répondre à ces besoins. Les applications ainsi développées sont des applications bureautiques pouvant être utilisées dans toute la Caisse Régionale Charente – Périgord, et des applications communautaires (utilisées dans toute la communauté Comète). Cette cellule se charge également de la mise à disposition des applications pour les utilisateurs et de l'éventuel dépannage à fournir en cas de problème.

2 Thème et activités

2.1 Présentation du sujet

Ce projet qui est destiné à une utilisatrice interne au crédit agricole, consiste à générer automatiquement des courriers pré remplis adressés aux personnes n'ayant toujours pas remboursé leur prêt in fine trois mois avant l'échéance. Ainsi, cela permettra d'avoir le moins de prêts possible impayés et qui seraient donc en état « d'anomalie ».

Le fait d'anticiper les règlements des prêts in fine va générer des économies de temps sur le traitement des dossiers NEIERTZ grâce aux effets conjoints de la réduction du nombre d'anomalie et de l'amélioration de l'information client.

Cette application devra être programmée en PHP et puis sera installée sur l'intranet du Crédit Agricole, afin que l'utilisatrice puisse y avoir accès.

2.2 Cahier des charges :

Contexte

Il s'agit de prêts sur des clients NEIERTZ. Ces crédits doivent être régularisés soit par la mise en place d'un nouveau plan BDF, soit par une rentrée prévue d'argent.

Cependant, il arrive fréquemment que les clients oublient l'arrivée à échéance de leur crédit et donc n'anticipent pas les modalités de son remboursement. Le prêt est alors identifié lors de son passage CDL, ce qui génère des dysfonctionnements pour sa régularisation.

Objectifs

- Faciliter la génération des courriers destinés à informer les clients NEIERTZ trois mois avant l'échéance de leurs prêts arrivés à terme, et leur rappeler les modalités de régularisation.

- Permettre l'édition mensuelle d'un courrier spécifique à adresser aux clients NEIERTZ qui ont un prêt in fine qui arrive à échéance dans 3 mois.

- Le courrier doit ressembler au modèle en **annexe1**

- Une gestion des courriers, ce qui permet de ne générer que les courriers nécessaires.

- Version imprimable d'une synthèse qui regroupe tout les noms des clients à qui nous envoyons une lettre.

- L'application doit être claire pour l'utilisatrice et facile d'emploi, mais le code se doit également être propre, en cas d'éventuelle modification a postériori.

Contraintes

Le projet doit être terminé à la fin de mon stage, il devra être validé par l'utilisatrice, puis contenir un manuel d'utilisateur.

2.3 Contact humain :

- Une réunion d'accueil avec toute l'équipe bureautique a été organisée par Stéphane Laurent afin de me présenter l'équipe bureautique, et me donner des informations sur le Crédit Agricole, ainsi que les autres Caisses Régionales du Crédit Agricole, membres du GIE COMETE.

- Nous avons contacté Mr Lascaud par téléphone, afin d'avoir plus d'information sur la base de données ACCESS qu'il nous a envoyée, plus précisément, sur la nature des données que nous y trouverons.

- J'ai demandé de l'aide sur PHP à Bernard Mayer. C'est lui qui me donne des conseils sur les logiciels constitutifs de mon environnement de travail, qui sont les plus pratiques à utiliser, et qui me conseille aussi sur les méthodes à appliquer.

- Nous avons contacté Mme Vallantin , l'utilisatrice, par courriel puis, plus tard, par téléphone afin de lui demander de tester l'application et de recueillir son opinion pour qu'elle nous précise la conformité aux attentes, et qu'elle nous suggére quelques améliorations à apporter à cette application de manière à ce que l'on puisse l'optimiser. Par exemple nous avons rajouté des fonctionnalités qui permettent de ne pas gérer certains courriers précisément désignés. Il est possible d'empêcher la génération de courriers de certaines personnes, si pour une raison quelconque ils n'ont pas besoin d'être avertis.

2.4 Moyens matériels et logiciels

Moyens logiciels :

- Eclipse (avec l'installation du module PHP-Eclipse) : Eclipse est un éditeur qui permet de développer en ayant un espace de travail qui est facile à ordonner, et muni d'une foule de fonctionnalités améliorant la productivité.

- Microsoft Visio : Pour faire un MCD. Cependant à la suite d'une demi-journée d'essais infructueux, nous avons opté pour un autre programme (DBDesigner) plus rapide car il manquait des bibliothèques relatives à la modélisation de l'application à développer.

- DB Designer : C'est un logiciel gratuit qui permet la construction de MPD, ce programme est capable d'importer des tables d'une base de données existante. Toutefois, vu qu'il n'existait pas de liens nommément décris dans la base de données, il n'a pas été capable de les deviner.

- XAMPP : Permet d'installer un serveur Apache, le langage PHP, un moteur de base de données MySql, et d'autres outils annexes, sur un ordinateur. Cependant, après un bon moment, nous avons eu beaucoup de problèmes pour utiliser les fonctions de PDO (PDO sera décrit plus tard). L'équipe bureautique ne pouvant trop consacré de temps à diagnostiquer le pourquoi, contourna le problème en me faisant installer indépendamment chacun des composants nécessaires.

- MySQL administrator : Permet de se connecter à un serveur PHP et de l'administrer, c'est-à-dire que l'on peut changer la configuration du serveur, ou créer des utilisateurs et leurs droits sur une base de données.

- MySQL Query Browser : Permet de gérer les bases de données. A partir de ce programme nous pouvons, créer, modifier, supprimer des bases de données ainsi que leurs tables et leurs attributs, et, surtout, de manipuler les données contenues dans la base.

CRÉDIT AGRICOLE

- MySQL Workbench : Génère un MCD à partir d'une base de données que l'on a créée sous MySQL Query Browser. Ce produit est en réalité l'évolution de DBDesigner. Je le cite pour mémoire.

Moyens matériels :

- Cours de BD (PHP et SQL)

- Un serveur de développement et de tests monté par un développeur de l'équipe bureautique. Je m'en suis servi pour stocker ma base de données ainsi que pour tester mes pages PHP. Ce serveur possède le programme APACHE avec PHP 5 pour pouvoir tester les pages PHP et MySQL Server 5 pour héberger des bases de données MySQL.

- Imprimante (ce qui permet d'imprimer les manuels trouvés sur internet)

- Un ordinateur à ma libre utilisation

- Le dossier d'expression des besoins

CRÉDIT AGRICOLE

2.5 Méthodes de travail :

Lorsqu'une personne fait une demande de projet, un informaticien du service commence à étudier le problème afin de voir s'il est réalisable, puis il réécrit une partie du cahier des charges de manière plus technique afin que le besoin soit bien compris plus tard, lors d'une éventuelle évolution.
Le travail a commencé par la réception du cahier des charges accompagné d'une base de données ACCESS, que M Lascaud m'a envoyé.

- Pour entamer le projet, il a d'abord fallu comprendre le travail d'extraction de données réalisé par M Lascaud, ce qui va nous permettre de déterminer sur quelles données de la base nous allons nous appuyer pour générer les courriers.

- Ensuite nous établirons un MPD à partir des éléments de l'entrepôt de données (un autre serveur de bases de données) puis nous commencerons à construire la base de données MySQL à l'aide de « MySQL Query Browser ».

- Nous avons choisi le PHP car beaucoup d'applications sont construites à partir de cette technologie afin d'être accessibles facilement par l'intranet du Crédit Agricole. Pour avoir une interface agréable et une organisation des dossiers facilitée, j'utiliserais le logiciel « Eclipse ».

- Démonstration du programme à l'utilisatrice, et tests par celle-ci.

- Correction du programme (itération) jusqu'à la validation du projet par l'utilisatrice.

- Construction du manuel utilisateur

3 Développement de l'application :

3.1 L'analyse de la base de données ACCESS :

Microsoft Access est un Système de Gestion de Base de Données (SGBD) relationnelle. Il offre un ensemble d'outils permettant de saisir, de manipuler, d'interroger et d'imprimer des données.

La base de données ACCESS origine des données est composée de 3 tables remplies par 3 requêtes SQL depuis l'entrepôt de données. Vous pourrez voir en Annexe2 un exemple d'une de ces requêtes que je vous ai commenté en vert.

J'ai du rechercher les contraintes logiques entre les tables, et découvrir le pourquoi de chaque table, afin de comprendre leur raison d'être.

Le schéma ci-dessous représente les trois tables sélectionnées comme contenant des données viables et nécessaires.

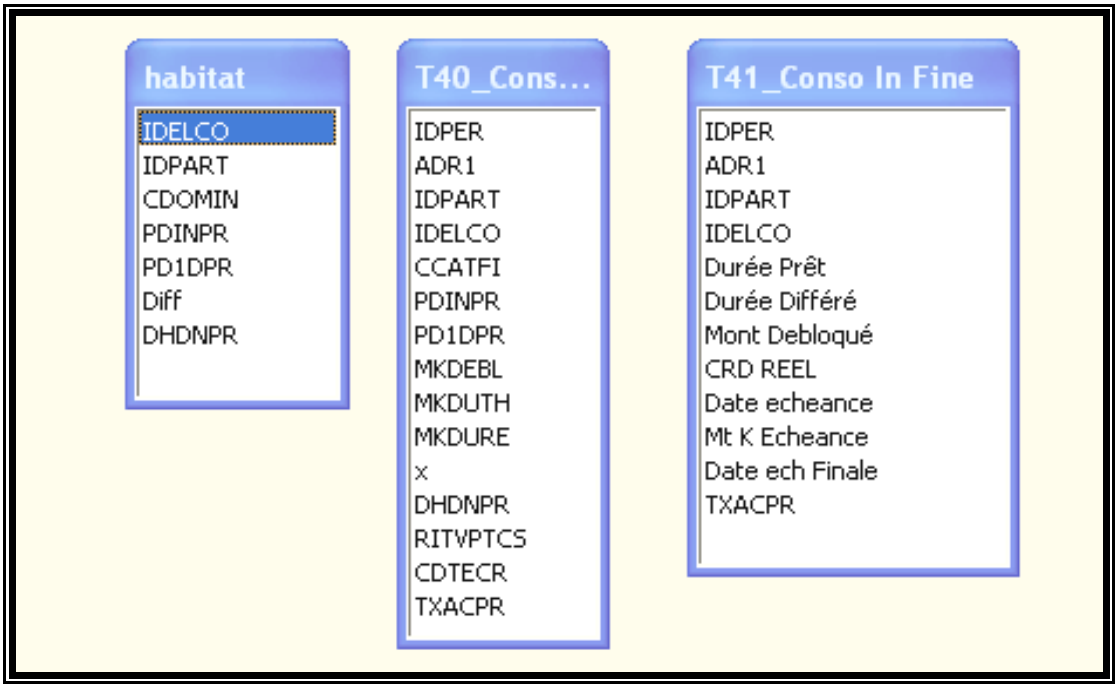


Schéma relationnel de la base de données ACCESS

CRÉDIT AGRICOLE

Cette base de données possède des liens vers les tables de l'entrepôt de données pour pouvoir récupérer les informations. Ces liens nous permettent de comprendre comment était construit l'entrepôt de données, cependant ceci n'en est qu'un extrait.

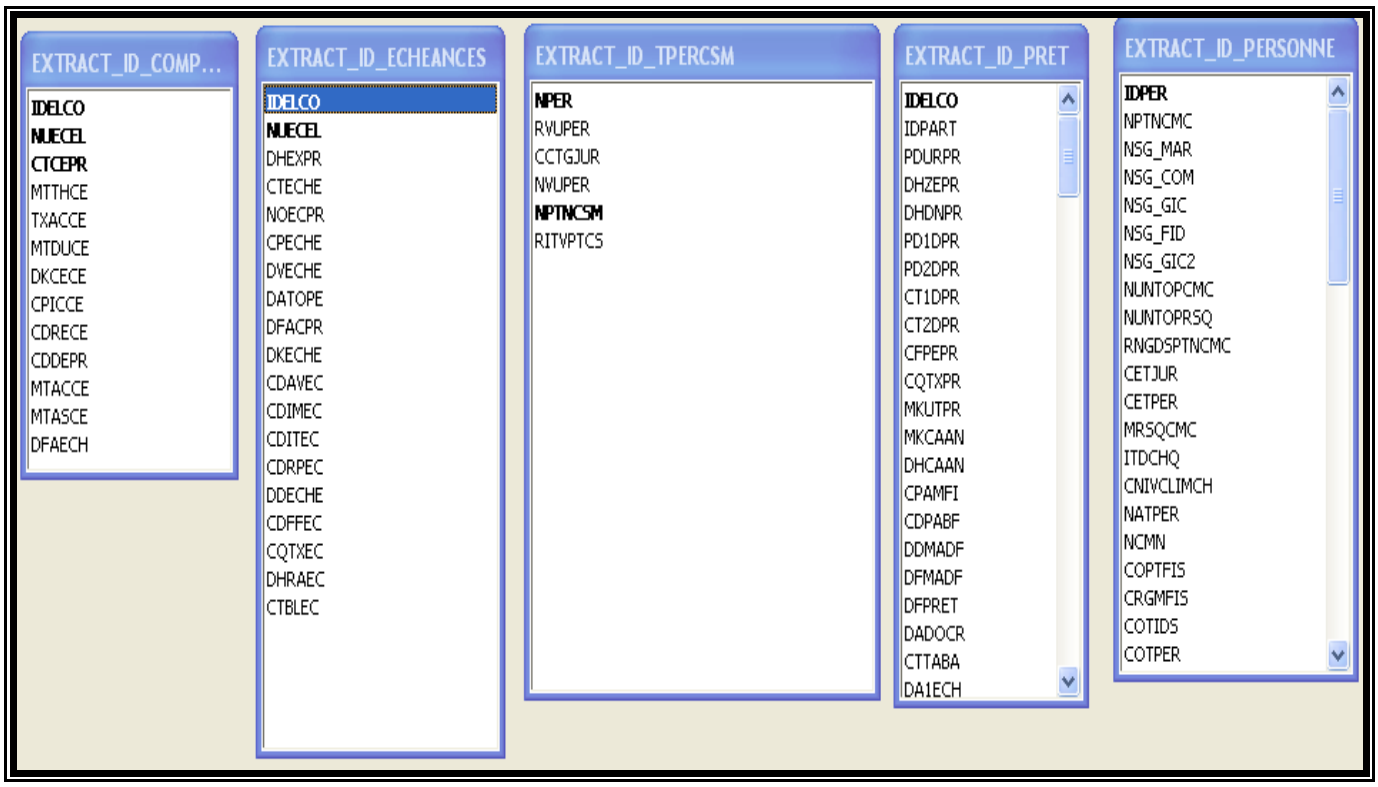
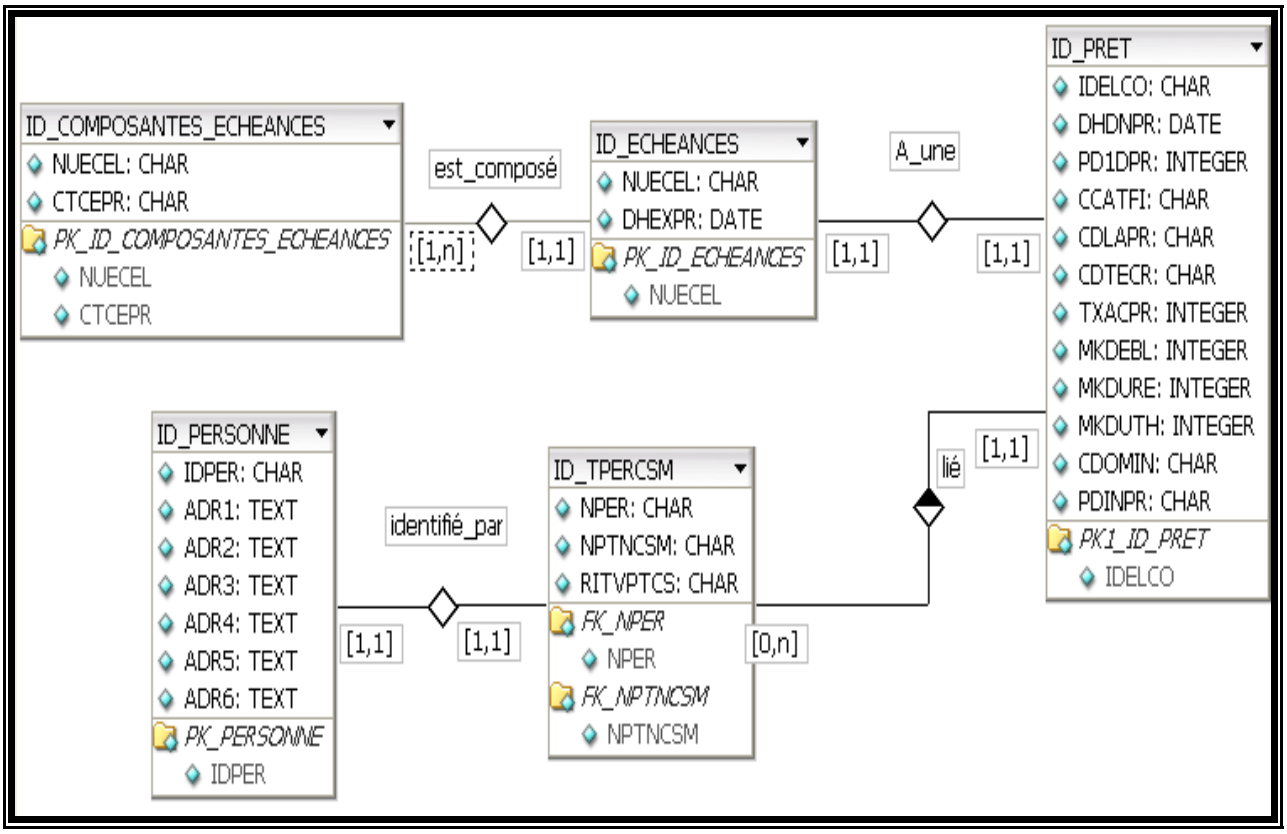


Schéma des tables de la base de données ACCESS

Etant donné la taille des tables je ne peux en montrer qu'une partie. Pour analyser le problème j'ai dû retrouver tous les attributs utilisés par les requêtes, et, muni de ceux-ci, créer un MPD de l'entrepôt de données avec juste les attributs que nous retrouvions dans les requêtes de la base ACCESS initiale.

3.2 Le MCD : Modèle Conceptuel de Données :

Tout d'abord j'ai réalisé le MCD des tables utilisées de l'entrepôt de données avec seulement les attributs qui vont servir de source, pour alimenter les tables de ma base de



données.

MCD de la Data Ware House

Description: Nous pouvons lire ce MCD de la manière suivante, une personne est identifiée par certains critères, critères qui sont liés à un ou plusieurs prêts. Ces prêts possèdent des échéances qui sont classées par certains critères.

Table <ID_PERSONNE>

Contient les renseignements sur une personne.

Table <ID_TPERCSM>

Contient des précisions sur une catégorie de personnes.

Table <ID_PRET>

Contient les détails des prêts.

Table <ID_ ECHEANCES >

Contient les dates d'échéances.

Table <ID_COMPOSANTE_ECHEANCES>

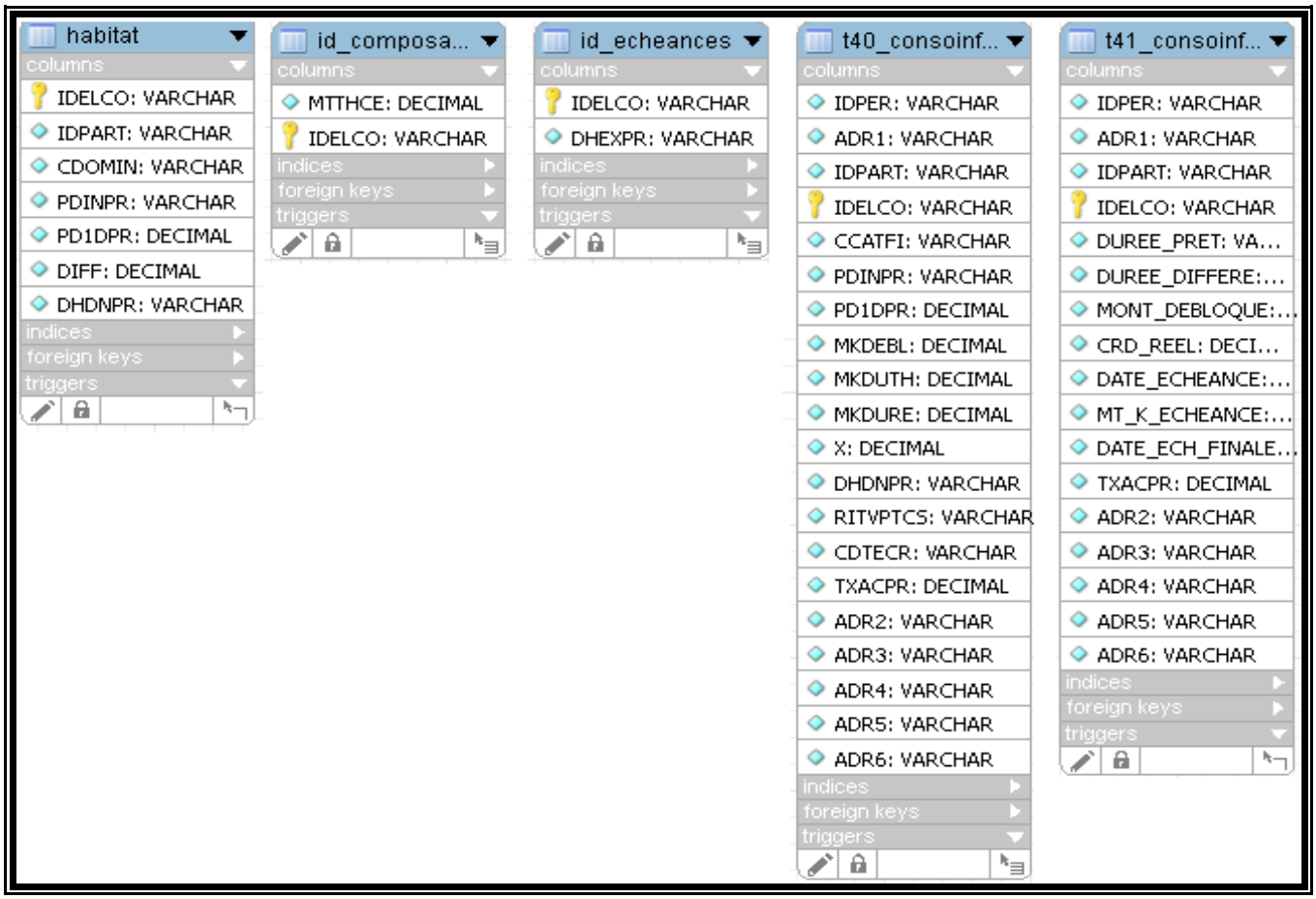
Contient des normes sur l'échéance.

Note : Ce MCD permet de schématiser la récupération des personnes avec leurs prêts, ainsi que les informations nécessaires pour la réalisation du courrier.

CRÉDIT AGRICOLE

Ensuite, j'ai fait un MCD pour la base de données cible que j'ai crée et qui s'inspire de la base de données ACCESS initiale. L'image de ce MCD est différente, car je l'ai généré avec «MySQL Workbench ». Ce logiciel permet de facilement générer un MCD provenant d'une base de données créée sous MYSQL Query Browser car ce sont des programmes initialement prévus pour fonctionner de concert.

MCD de la Base de données MySQL créée:



CRÉDIT AGRICOLE

Table <Habitat>

Cette table nous a amené à nous poser bien des questions tout le long du projet, car nous ne savions pas à quoi elle servait. En effet, la requête qui remplit cette table retourne en permanence un ensemble vide. Nous avons donc décidé de la recréer sans la toucher, au cas où elle serait utile plus tard.

Table <ID_COMPOSANTES_ECHEANCES>

Contient des précisions sur une catégorie de personnes.

Table <ID_ECHEANCES>

Contient les détails des prêts.

Table <t40_consoinfine >

Cette table est une table intermédiaire qui prépare les données nécessaires pour remplir « T41_consoinfine ». Elle regroupe certaines données qui viennent de l'entrepôt de données.

Elle sera comparée avec les tables <ID_ECHEANCES> et <ID_COMPOSANTES_ECHEANCES> suivant des critères pour former à la fin la table <T41_consoinfine>

Table <t41_consoinfine>

Cette table contient le résultat final, c'est à dire les clients avec leur(s) prêt(s), les dates et les montants nécessaires.

Note : En réalité, il n'y a pas de lien entre ces tables, car les autres tables ne sont que des intermédiaires pour récupérer les informations indispensables à l'application. Cela permet de pouvoir suivre ce qui se passe en cas de problèmes dans le remplissage des tables. Ce système de tables intermédiaires autorise le chargement des données de manière plus précise et ainsi de charger moins de lignes, ce qui procure en retour une meilleure vitesse d'exécution.

3.3 Gérer les bases de données sous les outils MySQL

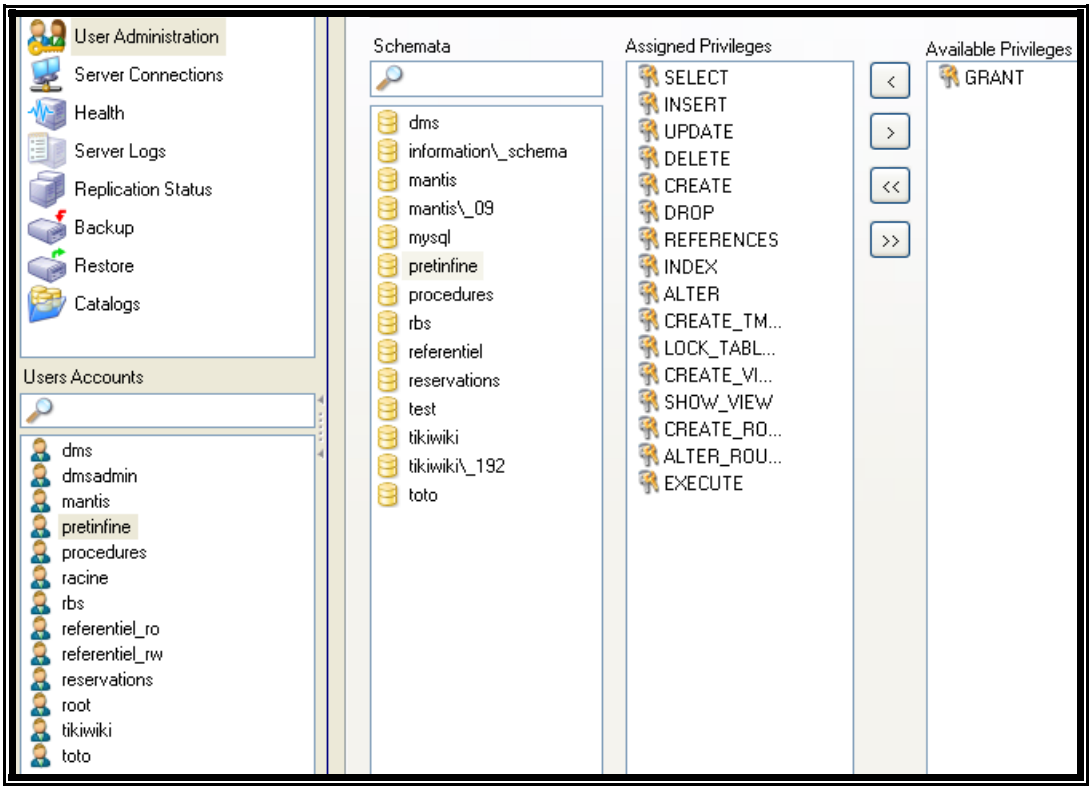
1.1.1 Présentation des outils MySQL

J'ai utilisé 3 outils MySQL permettant de gérer les bases de données:

- MySQL administrator pour gérer les droits des bases de données et certains paramètres de la base de données.
- MySQL Query Browser qui permet de créer de nouvelles bases de données, créer ou modifier des tables, et gérer leurs attributs, manipuler les données.
- MySQL Workbench qui permet de créer un MPD facilement à partir d'une base de données existante.

1.1.2 Création d'un utilisateur avec MySQL administrator :

J'ai dû créer un utilisateur que j'utiliserai pour faire une connexion à ma base de données. Pour cela j'ai créé un utilisateur Pretinfine et je lui ai donné tout les droits sauf le GRANT (c'est le droit de modifier les droits) sur la base de données « pretinfine ».

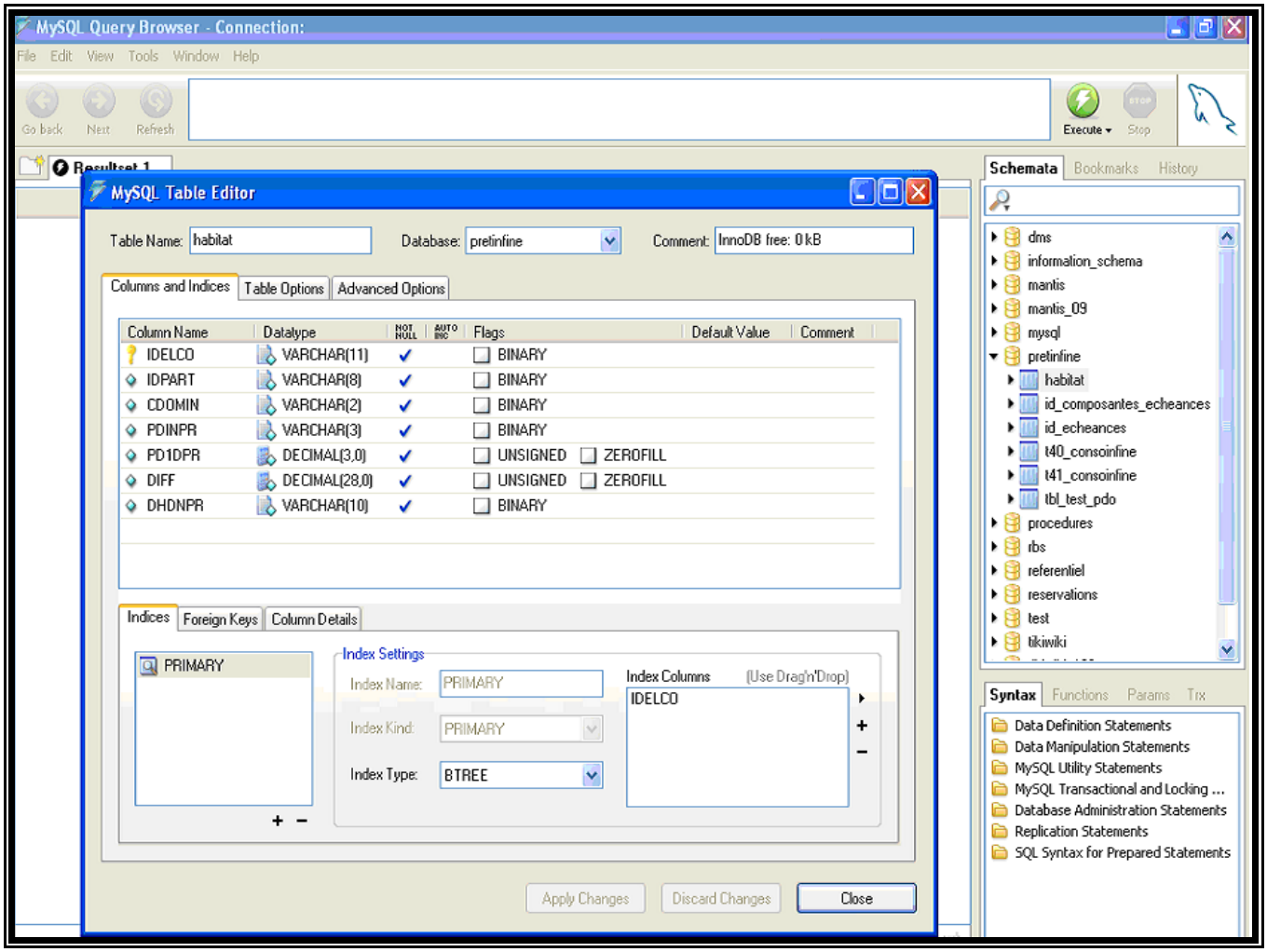


Capture d'écran : Gestion d'association des droits sous MySQL administrator:

CRÉDIT AGRICOLE

1.1.3 Création de la base de données sous MySQL Query Browser.

Pour créer la base de données sur le serveur de test, nous allons utiliser MySQL Query Browser. MySql Administrator est également capable de créer une base de données.
C'est un outil très simple d'utilisation, une fois dans le programme il a suffi de créer un nouveau dossier (pour créer la base de données). Ensuite il ne reste plus qu'à créer les tables, définir leurs attributs et le type des attributs (caractères, numérique).

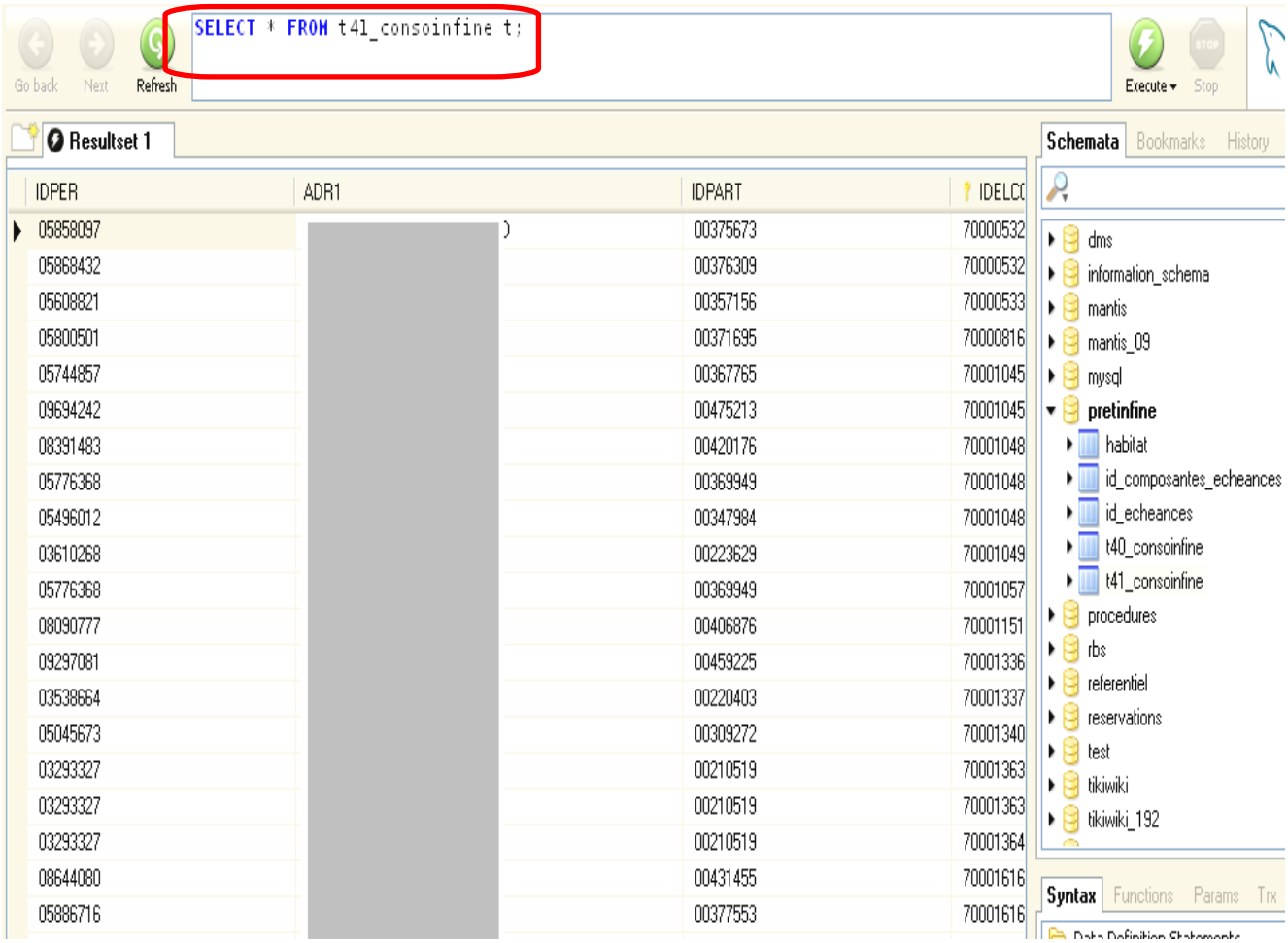


Capture d'écran :Création de table sur MySQL Query Browser:

CRÉDIT AGRICOLE

Ce même outil permet aussi d'effectuer des requêtes SQL. Cependant, son à la détection d'erreurs se limite à renvoyer au manuel ....

Voici un exemple d'utilisation de la requête dans le programme en question (les noms sont cachés pour des raisons de confidentialité) :



MCD de la Base de données MySQL créée:

3.4 La phase développement PHP :

1.1.4 Présentation de PHP

Le PHP est un langage de script, créé à l'origine en 1994 par Rasmus Lerdorf, principalement utilisé pour être exécuté par un serveur HTTP. Il peut également être utilisé comme n'importe quel autre langage si l'on possède l'interpréteur PHP sur sa machine. PHP permet de développer des scripts suivant le modèle procédural et/ou le modèle objet. On désigne parfois PHP comme une plate-forme plus que comme un simple langage.

L'utilisation la plus répandue est celle de générateur de page Web (contenant du code HTML, CSS, JavaScript, etc.) Ce langage permet donc de générer des pages au contenu dynamique (différent en fonction des données passées en paramètre, souvent par le biais de formulaires).

1.1.5 La méthode de développement :

Pour développer cette application je vais utiliser Eclipse comme logiciel.
Je vais remplir la base de données avec les personnes qui doivent être averties par courrier, c'est-à-dire les personnes n'ayant pas payé leur prêt 3 mois avant l'échéance.

- Pour commencer je dois établir des connexions aux bases de données, pour cela je suis astreint d'utiliser une extension de PHP (appelé PDO) qui permet de travailler sur 2 bases de données de structures différentes (ici on a du MySQL et du Oracle) sans faire de différence dans la manière de les traiter. L'abstraction permise par cette bibliothèque permet même d'implémenter des fonctionnalités à des bases de données alors qu'elles n'en disposent pas de façon native.

- Ensuite je reconstruis les requêtes ACCESS en SQL sur la plate-forme PHP. Les requêtes étant plus lentes, j'utiliserai l'extension de PHP (PDO) qui permet aussi de créer des requêtes préparées, méthode qui économisent les ressources de la machine et amènent un gain de temps.

- Je génère les courriers sous forme de fichiers PDF

CRÉDIT AGRICOLE

- Je construis l'interface afin que l'application soit facile à manipuler pour l'utilisateur.

1.1.6 Le PHP avec Eclipse :

Eclipse permet à l' origine de programmer avec des langages tels que JAVA ou C/C++. Cependant il est possible en installant un plugin de programmer en PHP, permettant ainsi de nous aider en mettant des couleurs et en recherchant des erreurs mineures (un oubli de parenthèse ou un oubli de ' ;' à la fin des phrases). De plus la hiérarchisation des dossiers est facile à manipuler ce qui fait gagner beaucoup de temps.

1.1.7 La programmation PHP avec PDO (PHP Data Objects):

PDO est une interface d'accès aux bases de données. Cette couche logicielle intégrable à PHP, permet d'accéder directement et de façon uniforme à n'importe quelle base de données parmi celles supportées (MySQL, SQLite, Oracle, …) sans se préoccuper du moteur qui tourne derrière. Ainsi, le jour où vous décidez de migrer vers une autre base de données, il vous suffit de modifier la déclaration de connexion à la base de données (sous forme d'URL) pour que votre application fonctionne automatiquement, sans avoir à toucher à votre code.
PDO permet aussi d'utiliser les requêtes de façon plus ou moins avancée (avec les requêtes préparées par exemple).

Pour un PHP 5.1 et supérieur il est simple d'activer PDO, il suffit d'ouvrir le fichier php.ini et puis de trouver les lignes concernant le PDO (exemple : « extension=php_pdo_mysql.dll ») et d'enlever le ' ;' qui déclare la ligne comme étant un commentaire.

CRÉDIT AGRICOLE

Voici un exemple de requête préparée que j'ai réalisée :

Requête préparée

```
//on définit la requête
$stmt = $cnxp->prepare("INSERT INTO id_echeances  (IDELCO,DHEXPR)
        VALUES (:idelco,:dhexpr)");

//on définit les variables de la requête préparé »
$stmt->bindParam(':idelco',$idelco);
$stmt->bindParam(':dhexpr',$dhexpr);


foreach ($rs as $row)           //$rs étant le résultat d'une requête
                                //avec un grand nombre de ligne
{
      //on met la date au bon format
      list($a,$m,$j)=preg_split('/\D+/',$row['DHEXPR']);
      $dhexpr = $j."/".$m."/".$a;
      $idelco = $row["IDELCO"];

      //on lance la requête
      $stmt->execute();
}
```

Avantage : La requête préparée est analysée une seule fois, mais sera exécutée plusieurs fois avec des paramètres identiques ou différents. Cela permet un gain de temps considérable lorsqu'on doit répéter l'analyse syntaxique plus d'une centaine de fois.

Inconvénient : Lorsque la requête est préparée, la base de données va analyser, compiler et optimiser son plan pour exécuter la requête. Ce qui peut rendre le procédé long pour des requêtes complexes.

Note : Il s'est avéré que les requêtes préparées ne mettent pas plus de temps pour n'être exécutées qu'une seule fois, que les requêtes ordinaires. Le gain est manifeste pour les requêtes suivantes.

1.1.8 Créer des fichiers PDF avec le langage PHP :

J'ai appris que l'on pouvait construire des fichiers PDF avec PHP. Alors, pour cela j'ai dû comprendre le fonctionnement et les fonctions de la bibliothèque PDF utilisée.
Voici les quelques fonctions de base retenues :

```
$pdf = new FPDF();    //Permet de créer un fichier PDF
$pdf->open();         //Ouvre le fichier PDF
$pdf->addPage();      //Créer une nouvelle page PDF
$pdf->SetX(20);       //Position de la zone à écrire
$pdf->SetFont('Times', 'B', 12);      //choix de la police

//Ecrit la cellule avec le texte et sa forme (encadré..)
$pdf->MultiCell(175, 5, 'Bonjour', $bordure, $crlf);
```

Fonctions PDF de base

J'ai donc réalisé les courriers sous le format demandé dans le cahier des charges.

MME [redacted]

DFI/RECOUVREMENT AMIABLE
N/Réf: GO/MV-JMT 0545204890
0545204857

Objet: DOSSIER BDF

Soyaux, le 30/04/2007

Madame,

Suite à votre plan Banque de France, un réaménagement de vos dettes a été réalisé.
Vous aviez bénéficié d'un report de dettes de 4232.23 €, qui arrive à échéance le 10/07/2008. Cette somme deviendra exigible à cette date.
C'est pourquoi nous vous informons, à titre préventif, de bien vouloir prendre contact avec la Banque de France, afin de réexaminer votre situation ou d'envisager le paiement de l'intégralité de la dette à l'échéance.
Etant à votre disposition pour tout renseignement complémentaire, et vous souhaitant bonne réception de la présente,

Nous vous prions d'agréer, Madame, l'assurance de nos salutations distinguées.

Le Responsable du Service

Capture d'écran : Exemple d'une lettre générée

CRÉDIT AGRICOLE

1.1.9 Utilisation d'un fichier de configuration .INI :

Afin de ranger les variables similaires entre les pages, et aussi de pouvoir récupérer certaines données à tout moment dans le programme, j'ai utilisé un fichier .INI qui permet d'enregistrer des variables avec leurs valeurs.

Ce paramétrage hors du code PHP permet d'être plus lisible que du langage de programmation et ainsi il peut être modifié directement par l'utilisateur, que ce soit directement avec un éditeur de texte, ou par le biais d'une page PHP pour le faire.

Voici la forme du fichier de configuration que j'ai créé :

```
entete_lettre = "images/entete.JPG"
pied_lettre = "images/pied.JPG"
bordure = "1"
crlf = "1"
mois = "2"


[db_pretinfine]
nom = "pretinfine"
host = "localhost"
user = "pretinfine"
pwd = "pretinfine"
```

Le fichier de configuration

CRÉDIT AGRICOLE

Voici comment peut être utilisé le fichier INI dans un programme :

```
//Ceci fait appel au ficher de configuration
//il est nécessaire de mettre au début du PHP
//si l'on veut utiliser le fichier INI
$config = parse_ini_file('config.ini',true);

//appel de la variable 'mois' depuis le fichier de
//configuration
$config['mois'] ;

//fonction permettant d'écrire dans le fichier de
//configuration
write_ini_file('config.ini',$config);
```

Capture d'écran : Le fichier de configuration

Pour pouvoir utiliser la fonction « write_ini_file » j'ai du recopier une fonction trouvée sur ce site : http://www.nexen.net.

Cette fonction permet d'écrire dans un fichier, elle change la valeur d'une variable si elle existe déjà, et elle créé une nouvelle variable si la variable enregistrée n'existe pas.

CRÉDIT AGRICOLE

1.1.10Remarque et Problème:

- La gestion des dates :

J'ai passé beaucoup de temps sur les dates, car entre les deux bases de données, le format était différent. Par exemple sur la base de données Oracle, origine de l'entrepôt de données, j'avais un format comme ceci « AAAA - MM - JJ 00:00:00 ». Ce format était détruit par ODBC, qui le converti en champs de type caractères. Comme l'application n'utilise guère de calcul de dates, il fût collégialement décidé de stocker à nouveau en format caractères dans la base de données cible.

Pour cela j'ai du trouver un moyen de scinder la date en plusieurs parties, chaque parties affectée à une variable. Ce qui ne fut pas une chose simple car j'ai dû utiliser une fonction avec une expression rationnelle (qui permet de séparer les dates sur des critères remarquables au sein de la chaînes de caractères).

Une fois les variables séparées j'ai juste eu besoin d'utiliser les fonctions de création de dates vues en cours.

- La détection d'erreurs :

Le serveur possède l'extension XDebug. C'est un logiciel permettant de donner des informations sur les erreurs détectées, et plus encore. Ces erreurs sont classées dans un tableau qui présente l'endroit où elles se trouvent, présentées sous forme hiérarchique. Il donne aussi le temps qu'utilisent certaines fonctions pour s'exécuter ce qui nous permet de voir les endroits dans le code qui consomment du temps.
Les variables, les paramètres fournis lors de l'appel des fonctions peuvent aussi apparaître, suivant la configuration de XDebug que l'on détermine.

Vous pouvez voir un exemple en annexe 3.

CRÉDIT AGRICOLE

4 Réflexion :

4.1 Résultat, la théorie et la pratique :

J'ai utilisé une analyse proche de la méthode Merise, comme on l'a vu en cours. Ensuite, après avoir créé la base de données, j'ai pu tout de suite démarrer la programmation en elle même. C'est une méthode très proche de celle que l'on a pu utiliser lors du projet en base de données que l'on a eu en fin d'année.
La différence est que la base de données sur laquelle je travaille est énorme, par conséquent certaines habitudes de programmation simple qui permettaient d'avoir simplement des résultats ne sont plus suffisantes car elles augmentent considérablement les temps de réponse lorsque l'on travaille sur plusieurs milliers de lignes.
J'ai donc du faire preuve de réflexion afin de déterminer des méthodes certes peu orthodoxes, mais amenant des temps d'exécution plus court.

D'ailleurs, la première version de l'application, les requêtes prenaient plus d'une minute pour se terminer, alors qu'à la version finale, le temps fût diminué à moins de 20 secondes.
Il a suffit d'utiliser les requêtes préparées de PDO, rappelons le, optimisent la gestion des ressources lors du lancement de la requête. J'ai aussi dû ajouter des conditions supplémentaires aux requêtes, afin de travailler sur des résultats dotés du moins de lignes possibles et donc d'avoir un meilleur temps de réponse.

Pendant que je programmais j'étais constamment en train de rechercher des solutions dans des documentations afin de trouver des fonctions qui résolvent mes problèmes ou améliorent les performances. C'est comme ça que j'ai découvert beaucoup de fonctions que je ne connaissais pas à l'école ou que j'avais très peu utilisé. Cependant, les bases acquises en cours m'ont permis d'évoluer rapidement et ainsi j'ai pu comprendre sans trop de problème ces nouvelles fonctions.

CRÉDIT AGRICOLE

4.2 Le rôle de l'utilisatrice :

L'utilisatrice est la personne qui doit valider mon projet lorsqu'elle sera satisfaite de l'application. Pour mon cas c'est Mme Vallantin qui testera l'application.
Je l'ai contactée par téléphone et par courriel pour lui faire tester l'application.
Ensuite nous avons discuté et nous avons pu découvrir des possibilités supplémentaires pour que le programme remplisse encore mieux les besoins de l'utilisatrice. Ainsi, je sais mieux ce qu'elle veut, et je peux proposer des solutions.

Par exemple, au premier test, elle m'a demandé s'il était possible qu'une personne ayant ses prêts échus le même mois puisse avoir ses prêts regroupés dans la même lettre.
Elle m'a aussi demandé de lui générer une synthèse des clients à qui nous envoyons les lettres pour lui simplifier le travail.

Sont rôle est donc de me donner les informations et sa façon de voir le fonctionnement du programme, afin que je puisse l'améliorer et satisfaire ses besoins.

CRÉDIT AGRICOLE

5 Chronologie du stage

Semaine 1:

Découverte de l'entreprise

Installation et prise en main des logiciels utilisés

Première analyse du sujet

Semaine 2 :

Lecture de cours sur PDO

Analyse approfondie du sujet

Construction de la base de données

Semaines 3 à 5 :

Développement de l'application

Appel de l'utilisatrice pour le 1er teste

Conception du rapport en parallèle

Semaine 6 :

Finition de l'application

Conception du rapport en parallèle

Semaine 7 :

Conception du rapport

Appel de l'utilisatrice pour le 2ieme test et validation du projet

Correction de l'application

Semaine 8-9 :

Finition du rapport de stage.

Conception du manuel utilisateur de l'application Prêt in fine.

Reconstruction d'un cahier des charges

Correction sur l'application

CONCLUSION:

Suite aux dix semaines de stage au sein de la cellule bureautique du Crédit Agricole Charente Périgord, j'ai pu mener à terme l'application PHP.

Grâce à cette application, l'utilisatrice a rapidement connaissance des clients qui ont l'échéance de leur prêt dans le mois recherché. Elle peut aussi gérer la liste des clients comme elle le souhaite, en supprimant des lignes du rapport d'exécution (vous pouvez voir le plan de l'application en annexe 4).

J'ai pris soin de mettre des commentaires afin que le programme soit le plus clair possible au cas où quelqu'un doive le reprendre ou l'enrichir. Donc ce produit peut être aisément modifié pour d'autres types de lettres, il suffit de changer les requêtes dans l'application, ensuite modifier le corps de lettre.

La phase de développement du projet m'a permis de me familiariser avec deux langages orientés web, le HTML et le PHP.

Ce stage m'a donné, en plus d'une expérience professionnelle, un aperçu positif de la vie en entreprise et m'a montré l'importance du travail en équipe. L'aspect relationnel avec le « client », ici l'utilisatrice, a lui aussi, été primordial.

Glossaire

ACCESS : Microsoft Access est un Système de Gestion de Base de Données (SGBD) relationnelle. Il offre un ensemble d'outils permettant de saisir, de manipuler, d'interroger et d'imprimer des données.

Base de données : Une base de données est un ensemble structuré d'information. Elle peut contenir : des tables, des requêtes, des formulaires, des états.

B.D.F : Banque De France.

C.A.C.P. : Crédit Agricole Charente Périgord.

C.A.S.A. : Crédit Agricole S.A.

C.D.L. : Client Douteux Litigieux.

C.R. : Caisse Régionale.

D.W.H.: DataWare House ou entrepôt de données. C'est le serveur de bases de données du Crédit Agricole qui est utilisé par les applications qui ne nécessite pas une mise à jour en temps réel. Celle-ci est faite chaque nuit, d'après des informations extraites de l'ordinateur central. Ainsi les utilisateurs sont répartis sur les bases de données et évite les saturations.

D.I.L. Direction Informatique et Logistique.

G.I.E. : Groupement d'intérêts économiques. Des entreprises se regroupent, afin de mettre des moyens en commun, de mutualiser des outils.

M.C.D. : Modèle conceptuel de données.

M.P.D. : Modèle physique de données

MySQL : C'est un célèbre fournisseur de base de données, dont le modèle économique est particulier, car cette entreprise vit de la vente de services, et non de la vente de ses produits (Logiciels Libres).

NEIERTZ : Dans le cadre du surendettement des familles et des particuliers, la loi Neiertz regroupe trois objectifs :

1. Développer la prévention du surendettement.
2. Organiser une procédure de règlement des problèmes liés au surendettement des familles et particuliers.
3. Responsabiliser davantage les prêteurs et les emprunteurs. Elle permet à ces personnes de pouvoir augmenter la durée de paiement de leurs dettes.

ODBC : Open DataBase Connectivity. Il s'agit d'un format défini par Microsoft permettant la communication entre des clients bases de données fonctionnant sous Windows et les bases de données extérieures.

PDO : PHP Data Objects.

CRÉDIT AGRICOLE

S.G.B.D. : Système de gestion de Base de Données.

S.Q.L.: Structured Query Langage. Langage structuré permettant de faire des requêtes sur des S.G.B.D.

S.I.: Système d'informations. Ensemble des outils informatiques nécessaires au fonctionnement de l'entreprise.

CRÉDIT AGRICOLE

Sources documentaires

Pour toutes les opérations sur PHP, je me suis aidé du manuel PHP trouver à l'adresse suivante : http://www.php.net/docs.php

Pour programmer en PHP sous éclipse je me suis servi du "`PHP Eclipse A User Guide`" fourni par le Crédit Agricole.

J'ai récupéré de la documentation les fonctions INI à l'adresse suivante :
http://www.nexen.net/articles/dossier/16901-le_mois_de_la_fonction_php_:_fichiers_de_configuration.php

Pour tous les problèmes HTML, je me suis aidé du site à l'adresse suivante :
http://geosurf.free.fr/docs/html/selfhtml/selfhtml.htm

J'ai appris à me servir de PDO sur : http://fr2.php.net/pdo

Pour créer des tableaux dans un fichier PDF j'ai regardé sur le site suivant :
http://jc-cornic.developpez.com/tutoriels/php/pdf/?page=page_3

Documentation de Xdebug :
http://files.derickrethans.nl/xdebug-vancouver7.pdf

Annexe 1 : Modèle de lettre à accomplir :

M.xxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxx

DFI/RECOUVREMENT AMIABLE
N/Réf: GO/MV-JMT ☎ 0545204890
☎ 0545204857

Objet. : DOSSIER BDF

Soyaux, le 99/99/9999

Madame, Monsieur,
Suite à votre plan Banque de France, un réaménagement de vos dettes a été réalisé. Vous aviez bénéficié d'un report de dettes de xxxx €, qui arrive à échéance le 99/99/9999. Cette somme deviendra exigible à cette date.
C'est pourquoi nous vous informons à titre préventif de bien vouloir prendre contact avec la Banque de France afin de réexaminer votre situation ou d'envisager le paiement de l'intégralité de la dette à l'échéance.
Etant à votre disposition pour tout renseignement complémentaire, et vous souhaitant bonne réception de la présente,
Nous vous prions d'agréer, Madame, Monsieur, l'assurance de nos salutations distinguées.

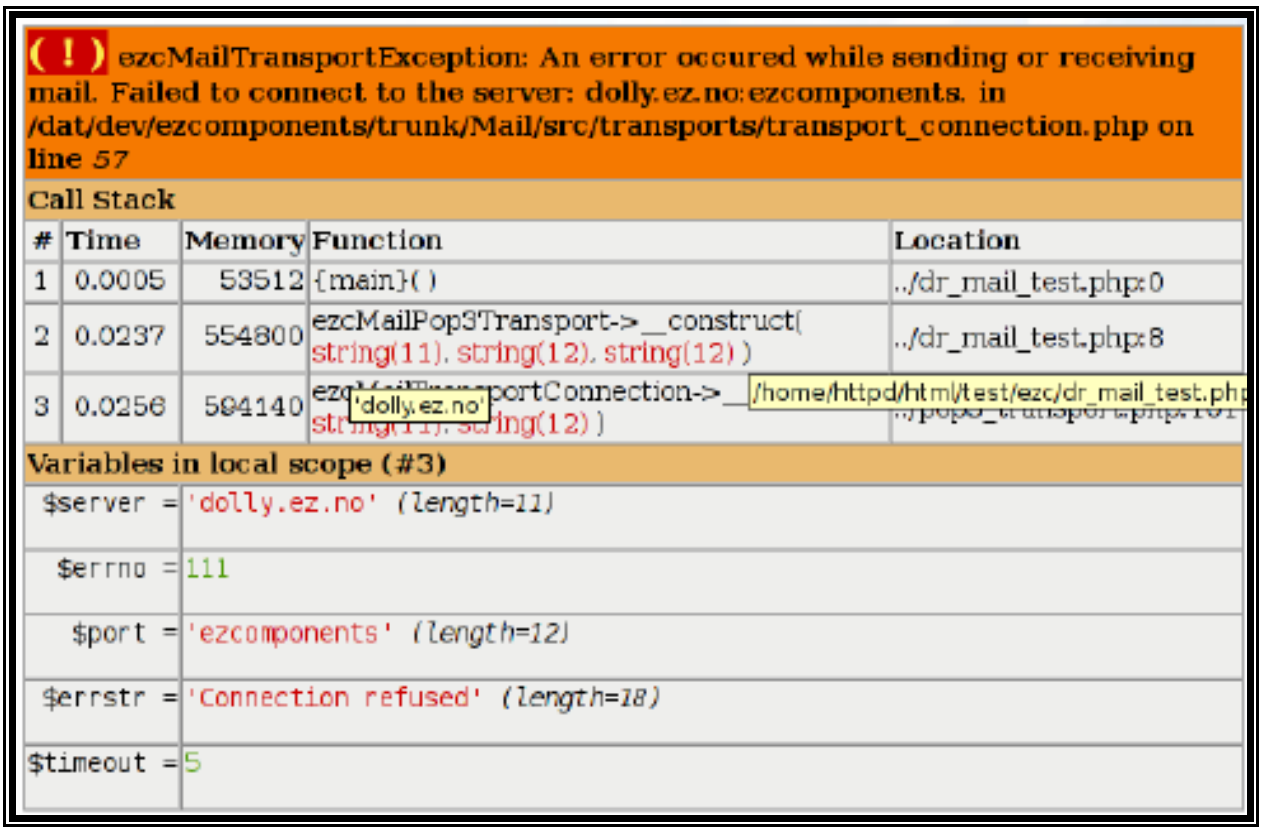
Le Responsable du Service

Annexe 2 : Exemple d'une requête SQL :

```
In fine Conso R40:
                //Les variables que je dois retenir
SELECT
EXTRACT_ID_PERSONNE.IDPER, EXTRACT_ID_PERSONNE.ADR1,
EXTRACT_ID_PRET.IDPART, EXTRACT_ID_PRET.IDELCO,
EXTRACT_ID_PRET.CCATFI, EXTRACT_ID_PRET.PDINPR,
EXTRACT_ID_PRET.PD1DPR, EXTRACT_ID_PRET.MKDEBL,
EXTRACT_ID_PRET.MKDUTH, EXTRACT_ID_PRET.MKDURE,
([mkduth]=[mkdure]) AS x,EXTRACT_ID_PRET.DHDNPR,
EXTRACT_ID_TPERCSM.RITVPTCS,EXTRACT_ID_PRET.CDTECR,
EXTRACT_ID_PRET.TXACPR INTO [T40_Conso In Fine]
                //Les table que l'on extrait
FROM EXTRACT_ID_PRET INNER JOIN (

      EXTRACT_ID_TPERCSM INNER JOIN EXTRACT_ID_PERSONNE

      ON EXTRACT_ID_TPERCSM.NPER = EXTRACT_ID_PERSONNE.IDPER
                                              //un lien entre 2


     )
     ON EXTRACT_ID_PRET.IDPART = EXTRACT_ID_TPERCSM.NPTNCSM
                //Les conditions
WHERE (
     (
      (EXTRACT_ID_PRET.CCATFI) = "05"
        Or (EXTRACT_ID_PRET.CCATFI) = "07"
        Or (EXTRACT_ID_PRET.CCATFI) = "11"
         )
     AND (
           (EXTRACT_ID_PRET.DHDNPR) > //recupere le mois prochain
DateSerial(Year(Now()),Month(Now()) + 1,1)
          And (EXTRACT_ID_PRET.DHDNPR) <
DateSerial(Year(Now()),Month(Now()) + 2,0)
           )
        AND ((EXTRACT_ID_TPERCSM.RITVPTCS) = "01")
        AND ((EXTRACT_ID_PRET.CDTECR) <> "25")
        AND ((EXTRACT_ID_PRET.TXACPR) = 0)
        AND ((EXTRACT_ID_PRET.CDOMIN) = "40"));
```


CRÉDIT AGRICOLE

Annexe 3 : Exemple de détection d'erreur:



(!) ezcMailTransportException: An error occured while sending or receiving mail. Failed to connect to the server: dolly.ez.no:ezcomponents. in /dat/dev/ezcomponents/trunk/Mail/src/transports/transport_connection.php on line 57

Call Stack

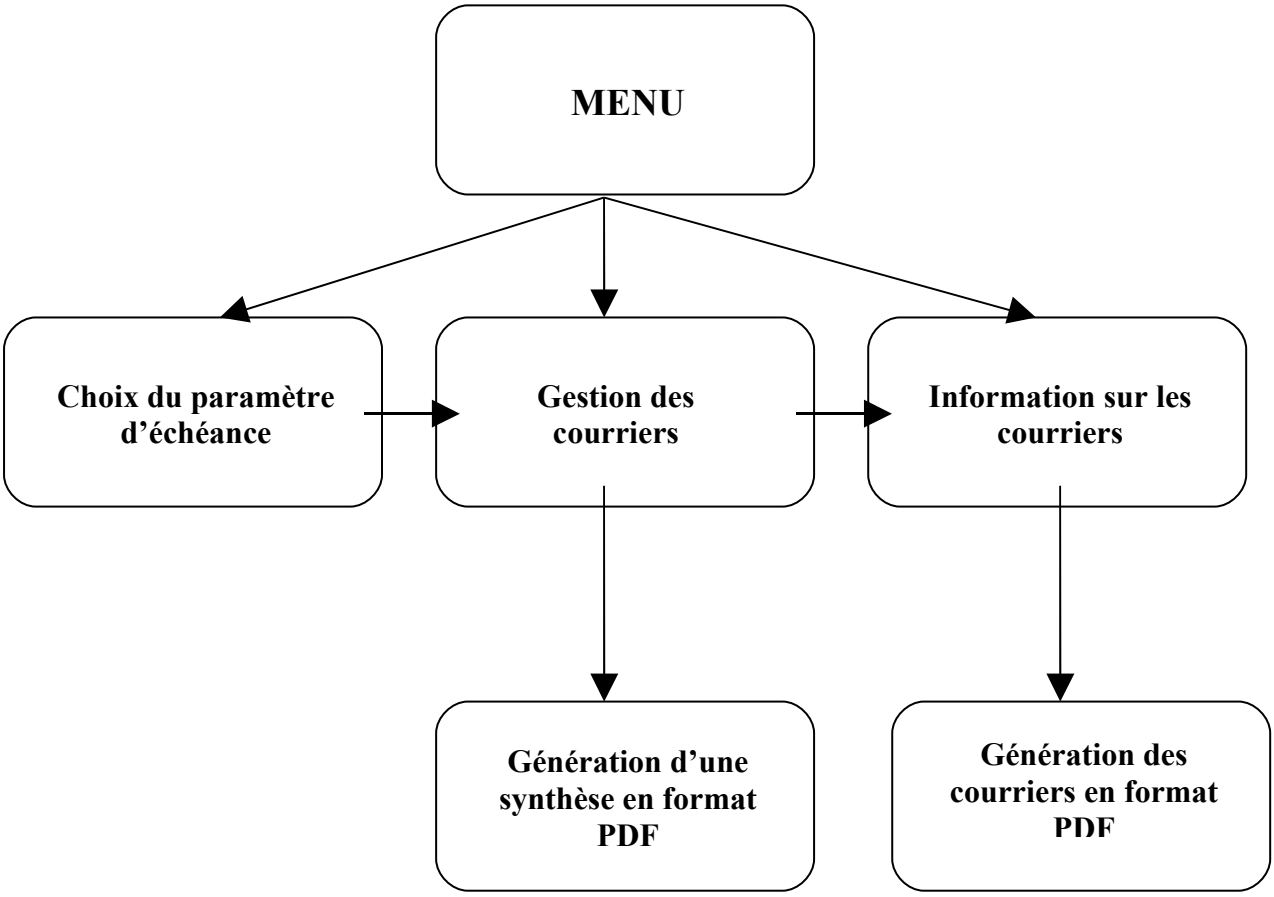
#	Time	Memory	Function	Location
1	0.0005	53512	{main}( )	../dr_mail_test.php:0
2	0.0237	554800	ezcMailPop3Transport->__construct(string(11), string(12), string(12))	../dr_mail_test.php:8
3	0.0256	594140	ezc...portConnection->__ 'dolly.ez.no' /home/httpd/html/test/ezc/dr_mail_test.php string(11), string(12))	

Variables in local scope (#3)

$server =	'dolly.ez.no' (length=11)
$errno =	111
$port =	'ezcomponents' (length=12)
$errstr =	'Connection refused' (length=18)
$timeout =	5

CRÉDIT AGRICOLE

Annexe 4 : Plan de l'application


MENU
Choix du paramètre d'échéance
Gestion des courriers
Information sur les courriers
Génération d'une synthèse en format PDF
Génération des courriers en format PDF

CRÉDIT AGRICOLE